AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method for performing a process associated with a QoS-guaranteeing multi-path in a path-based communication network having a plurality of nodes, comprising the steps of:

a) determining a start node, a destination node and a requirement condition associated with a reference cost consumed in a range from the start node to the destination node;

b) creating a first tree adapting the start node as its root and a second tree adapting the destination node as its root, including nodes close to the start node in a first node group, and including nodes close to the destination node in a second node group;

c) selecting a node having a minimum cost associated with the roots from among a plurality of nodes contained in the first and second node groups, and including the selected node having the minimum cost in a tree of a corresponding root;

d) if the selected node included in the tree at the step (c) is also included in the first and second trees, and a cost consumed in the range from the start node to the destination node on the basis of the selected node is less than the reference cost, providing a corresponding path associated with the cost;

e) including a nearby node of the selected node in a node group having the selected node when the selected node included in the tree at the step (c) is also included in either one of the first and second trees, comparing two paths ranging from a root ~~(i.e., a root of a nearby node's tree)~~ to the nearby node when the nearby node is previously included in the node group, and deleting a link of a cheaper one of the two paths; and

f) determining whether there is a node contained in the first node group and the second node group, repeatedly performing a process from the step (c) when the node is found in the first and second node groups, or terminating the process from the step (c) when no node is found in the first and second node groups.

2. (Original) The method according to claim 1, further comprising the steps of:

g) determining a maximum number of paths satisfying the determined requirement condition; and

h) if a number of output paths is higher than the maximum number of paths, terminating a path setup program.

3. (Original) The method according to claim 1, wherein the step (d) for outputting a path includes the steps of:

d1) determining whether the same path as a currently calculated path is previously outputted; and

d2) if the same path as the current path is not outputted, outputting a corresponding path equal to the current path.

4. (Original) The method according to claim 3, wherein the step (d1) includes the step of:

if it is determined that a node equal to another node "Si+1" is found in child nodes of output paths (i.e., S1, S2, . . . , Si, . . . , Sn) from a start root S1 to a root Si, determining that the same path as a path associated with the node "Si+1" has been previously outputted.

5. (Original) The method according to claim 3, wherein the step (d2) includes the steps of:

if it is determined that a node equal to another node "Si+1" is not found in son nodes of output paths (i.e., S1, S2, . . . , Si, . . . , Sn) from a start root S1 to a root Si, creating the node "Si+1" as a child node of the node Si and outputting the path associated with the node "Si+1".

6. (Original) The method as set forth in claim 1, further comprising the steps of:

i) selecting a first path from the start node to the destination node from among a plurality of paths outputted at the step (d); and

j) selecting at least one second path disjointed from an intermediate node existing in the first path from among multiple paths ranging from the start node to the destination node, and providing a disjoint path.

7. (Original) The method as set forth in claim 6, wherein the step (h) includes the steps of:

h1) determining whether the start node or the destination node has a degree of less than 2; and

h2) if the start node or the destination node has a degree of less than 2, determining that there is no disjoint path.

8. (Original) The method as set forth in claim 1, further comprising the steps of:

k) selecting two paths from among a plurality of paths outputted at the step (d); and

l) determining whether the selected two paths are disjointed from each other, and providing a disjoint path according the determination result.

9. (Original) The method as set forth in claim 8, wherein the step (l) includes the step of:

determining whether a nearby node of the start node of a first node contained in the two paths is equal to that of the destination node of a second node contained in the two paths, and determining a disjoint path according to the determination result.

10. (Currently Amended) A method for performing a process associated with a QoS-guaranteeing multi-path in a path-based communication network having a plurality of nodes, comprising the steps of:

a) determining a start node, a destination node and a requirement condition associated with a reference cost consumed in a range from the start node to the destination node;

b) creating a tree adapting an intermediate node as a root, and including nearby nodes close to the tree in a node group;

c) selecting one node having the lowest cost associated with the root from among nodes contained in the node group, and including the selected node in the tree of the root;

d) including a nearby node of the selected node included in the tree at the step (c) in the node group, comparing two paths ranging from a root ~~(i.e., a root of a nearby node's tree)~~ to the nearby node when the nearby node is previously included in the node group, and deleting a link of a cheaper one of the two paths; and

e) determining whether there is a node contained in the node group, repeatedly performing a process from the step (c) when the node is found in the node group, and outputting

a path from the intermediate node to the start node and the other path from the intermediate node to the destination node.

11. (Original) The method according to claim 10, further comprising the steps of:

f) determining a maximum number of paths satisfying the determined requirement condition; and

g) if a number of output paths is higher than the maximum number of paths, terminating a path setup program.

12. (Original) The method according to claim 10, wherein the step (e) for outputting a path includes the steps of:

e1) determining whether the same path as a currently calculated path is previously outputted; and

e2) if the same path as the current path is not outputted, outputting a corresponding path being the current path.

13. (Original) The method according to claim 12, wherein the step (e1) includes the step of:

if it is determined that a node equal to another node "Si+1" is found in child nodes of output paths (i.e., S1, S2, . . . , Si, . . . , Sn) from a start root S1 to a root Si, determining that the same path as a path associated with the node "Si+1" has been previously outputted.

14. (Original) The method according to claim 12, wherein the step (d2) includes the steps of:

if it is determined that a node equal to another node "Si+1" is not found in son nodes of output paths (i.e., S1, S2, . . . , Si, . . . , Sn) from a start root S1 to a root Si, creating the node "Si+1" as a child node of the node Si and outputting the path associated with the node "Si+1".

15. (Original) The method as set forth in claim 10, further comprising the steps of:

h) selecting a first path from the start node to the destination node from among a plurality of paths outputted at the step (e); and

i) selecting at least one second path disjointed from an intermediate node existing in the first path from among multiple paths ranging from the start node to the destination node, and providing a disjoint path.

16. (Original) The method as set forth in claim 15, wherein the step (i) includes the steps of:

i1) determining whether the start node or the destination node has a degree of less than 2; and

i2) if the start node or the destination node has a degree of less than 2, determining that there is no disjoint path.

17. (Original) The method as set forth in claim 10, further comprising the steps of:

j) selecting two paths from among a plurality of paths outputted at the step (d); and

k) determining whether the selected two paths are disjointed from each other, and providing a disjoint path according the determination result.

18. (Original) The method as set forth in claim 17, wherein the step (k) includes the step of:

determining whether a nearby node of the start node of a first node contained in the two paths is equal to that of the destination node of a second node contained in the two paths, and determining a disjoint path according to the determination result.

19. (Currently Amended) A computer-readable recording medium for storing a program in a processor, said program comprising the steps of:

determining a start node, a destination node and a requirement condition associated with a reference cost consumed in a range from the start node to the destination node;

b) creating a first tree adapting the start node as its root and a second tree adapting the destination node as its root, including nodes close to the start node in a first node group, and including nodes close to the destination node in a second node group;

c) selecting a node having a minimum cost associated with the roots from among a plurality of nodes contained in the first and second node groups, and including the selected node having the minimum cost in a tree of a corresponding root;

d) if the selected node included in the tree at the step (c) is also included in the first and second trees, and a cost consumed in the range from the start node to the destination node on the basis of the selected node is less than the reference cost, outputting a corresponding path associated with the cost;

e) including a nearby node of the selected node in a node group having the selected node when the selected node included in the tree at the step (c) is also included in either one of the first and second trees, comparing two paths ranging from a root ~~(i.e., a root of a nearby node's tree)~~ to the nearby node when the nearby node is previously included in the node group, and deleting a link of a cheaper one of the two paths; and

f) determining whether there is a node contained in the first node group and the second node group, repeatedly performing a process from the step (c) when the node is found in the first and second node groups, or terminating the process from the step (c) when no node is found in the first and second node groups.